

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

**Pixels editor**

propusă de

***Ana Maria-Daniela Condrea***

**Sesiunea:** *februarie, 2020*

Coordonator științific

**Lect. Dr. Anca Ignat**

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ

# **Pixels editor**

***Ana Maria-Daniela Condrea***

**Sesiunea:** *februarie, 2020*

Coordonator științific

***Lect. Dr. Anca Ignat***

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele \_\_\_\_\_

Data \_\_\_\_\_ Semnătura \_\_\_\_\_

### **DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul(a) .....

domiciliul în .....

născut(ă) la data de ....., identificat prin CNP .....,  
absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de  
..... specializarea ....., promoția  
....., declar pe propria răspundere, cunoscând consecințele falsului în  
declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr.  
1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

\_\_\_\_\_

\_\_\_\_\_elaborată sub îndrumarea dl. / d-na  
\_\_\_\_\_, pe care urmează să o susțină în fața  
comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin  
orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea  
conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări  
științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei

lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi, .....

Semnătură student .....

## DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Pixels editor*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, *data*

Absolvent *Ana Maria-Daniela Condrea*

---

(semnătura în original)

# 1.Cuprins

3. Contribuții.....	11
4. Implementare .....	12
4.1. Crearea meniului.....	12
4.1.1. Hide image .....	12
4.1.2. Hide text.....	14
4.1.2.1. Hide text invisible .....	14
4.1.2.2. Hide text visible .....	15
4.1.3. Hide audio .....	16
4.1.3.1. Hide audio in image .....	16
4.1.3.2. Hide image in audio .....	18
4.1.3.3. Hide text in audio.....	18
4.1.4. Hide video .....	19
4.1.4.1. Hide image in video .....	19
4.1.4.2. Hide text in video.....	20
4.1.5. Watermark .....	20
4.1.5.1. Add image logo to image .....	21
4.1.5.2. Add text logo to image.....	22
4.1.5.3. Add image logo to video .....	23
4.1.5.4. Add text logo to video.....	24
5. Testare .....	25
6.Concluzii.....	28
7. Bibliografie .....	29

## 2.Introducere

Întrucât spațiul online nu este sigur și există posibilitatea preluării datelor care nu se doresc a fi descifrate, s-au creat diferite modalități de a ascunde un mesaj fără ca cei din jur să observe acest lucru. Ascunderea unor informații prin criptografie presupune criptarea acestora prin utilizarea unor funcții Hash și chei publice care transforma mesajul într-unul vizibil întregii lumi. Prin steganografie mesajul ascuns este plasat într-un alt obiect, doar expeditorul și receptorul știu de existența sa.

Tema acestui proiect, ascunderea de mesaje în imagini, audio sau video, a reieșit din dorința de a crea o aplicație care reușeste să îmbine ascunderea de informații sub diferite forme. De asemenea am ales această temă deoarece am vrut să înțeleg cum pot descompune obiectele în pixeli și cum pot crea noi obiecte pe baza unor algoritmi care modifică doar câțiva biți din obiectul inițial.

Consider că această aplicație ajută la o mai bună înțelegere a conceptului de ascundere a unui mesaj. Reprezentarea grafică aduce un plus deoarece permite fiecărui utilizator să testeze atât modul prin care se ascund mesajele cât și modul prin care se extrag.

Ideea mea pentru acest proiect de licență este dezvoltarea unei aplicații desktop ce se poate utiliza de către persoanele care doresc să testeze modalități diferite de a ascunde o informație. Totodată ajută și la înțelegerea conceptului de steganografie întrucât pe lângă algoritmi de baza aduce noi elemente care permit execuția acestora pe mai multe cazuri. Aplicația permite utilizatorului să încarce imagini, text, fișiere audio sau video alături de mesajul pe care dorește să îl ascundă. După adăugarea tuturor parametrilor necesari se generează un nou obiect care va conține informația secretă. Receptorul fișierului rezultat anterior are posibilitatea de a prelua mesajul prin rularea comenzii de “unload”.

Steganografia(scriere ascunsă) reprezintă arta și știința de a scrie mesaje ascunse de așa maniera încât, în afara de expeditor și receptor, nimeni să nu poată suspecta și nici descifra existența mesajului ascuns. Steganografia reprezintă un domeniu larg, cu multiple forme de implementare și aplicabilitate, însă în ce privește securitatea datelor, steganografia se încadrează în categoria metodelor de *securizare prin obscuritate*.

## Steganografie și Criptografie

Diferența majoră între steganografie și criptografie constă în faptul că steganografia protejează mesajul prin ascunderea lui față de ceilalți cititori, față de criptografie care afișează întreaga informație criptată, și care, oricât de idescifrabilă ar fi, atrage atenția și curiozitatea asupra ei.

Steganografia	Criptografia
○ datele nu pot fi alterate, modificate în timpul transferului.	○ datele pot fi modificate
○ suportă principiile de confidențialitatea și autenticitate	○ suportă atât principiile de confidențialitate și autenticitate, precum și integritatea datelor
○ obiectivul principal este de a partaja un mesaj secret	○ principalul obiectiv este de a proteja datele partajate
○ se poate implementa pe audio, video, text sau imagini	○ se poate implementa doar pe text
○ mesajele sunt ascunse	○ mesajele sunt codificate

## Watermarking

Watermarking-ul, sau marcarea transparentă, este operația de introducere a unei informații transparente, imperceptibile pentru sistemul vizual numită marcaj watermark în informația gazdă, care poate fi text, semnal audio, imagine statică sau video. Steganografia reprezintă o formă de implementare a principiului de watermarking deoarece presupune ascunderea unor mesaje fără ca acestea să fie detectate.



Procesul de marcare transparentă pentru a fi cât mai corect și să realizeze protecția datelor trebuie să îndeplinească două condiții:

1. introducerea marcajelor în datele gazdă înainte ca acestea să fie partajate.
2. extragerea mesajelor din datele recepționate și compararea informației adăugate la emisie cu cea extrasă la recepție, pentru autentificare, în caz de atac asupra fișierului partajat.

O aplicație asemănătoare ar fi Image Steganography. Această aplicație este creată în JavaScript și are ca și funcționalitate ascunderea unei imagini într-o altă imagine cu un număr specificat de biți. De asemenea oferă și posibilitatea de a prelua mesajul ascuns din imaginea generată la pasul anterior. Totuși, există și câteva aspecte negative ale acestei aplicații, Image Steganography nu permite utilizatorilor să apeleze aceste funcționalități pe mai multe domenii precum video, text sau audio și se rezumă doar la ascunderea de imagine în imagine. Acest lucru nu permite o vastă înțelegere asupra conceptului de steganografie și limitează execuția programului doar la un singur element din cele patru medii de lucru pe care le permite această știință.

O altă aplicație similară este Steganography(encode text into image). Aplicația are ca și funcționalitate ascunderea de text sau de imagini într-o alta imagine. Steganography este mult mai amplă față de aplicația anterioară dar prezintă anumite restricții:

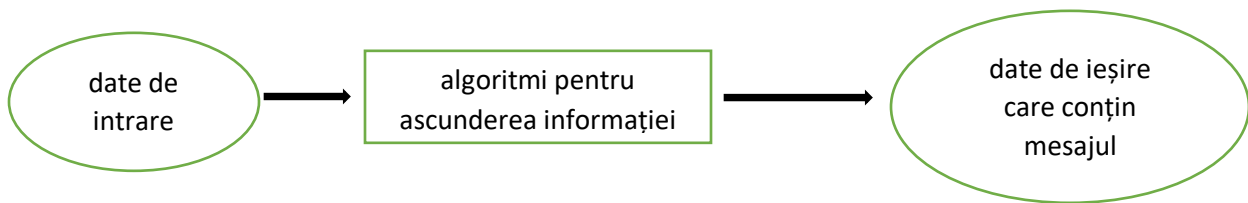
- nu putem alege câți biți vrem să modificăm;
- nu permite utilizarea unor imagini mari, robuste;
- nu poate fi folosit pentru ascunderea de informații în audio sau video;
- nu prezintă modul clar de a utiliza aplicația;

Diferența dintre proiectele existente în mediul online și proiectul meu, Pixels editor, îl reprezintă generalitatea acestuia pe toate cele patru medii de lucru ale steganografiei. Pe lângă funcționalitățile de ascundere și preluare de mesaje din imagini, audio sau video aplicația vine cu un plus, acesta fiind posibilitatea de a adăuga watermark în imagini sau video.

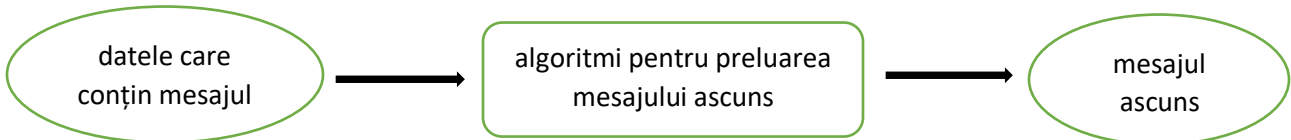
Modul de lucru al aplicației este simplu, întrucât există un meniu simplu cu submeniuri clare care permit utilizatorului să acceseze funcționalitățile aplicației cu ușurință. Flow-ul aplicației este unul clar, ușor de înțeles și constă în preluarea unor date de intrare setate de utilizator și în funcție de necesitate acesta poate ascunde sau prelua un mesaj.

Aplicația realizează două mari funcționalități:

1. Ascunderea de informații: utilizatorul completează cu date valide câmpurile cerute după care urmează apăsarea butonului de Load.



2. Preluarea informației ascunse: utilizatorul completează câmpurile cu datele rezultate la pasul de încărcare după care apasă butonul de Unload.



Atât pentru ascunderea de informații cât și pentru preluarea informației ascunse completarea câmpurilor este testată. Astfel fiecare componentă care trebuie completată este verificată în ideea corectitudinii contextului:

- căile introduse trebuie să fie valide;
- fiecare componentă adăugată manual trebuie să fie setată pentru a fi vizibilă în program;
- în cazul prezenței unui radio button trebuie neaparat să alegem o valoare;

### 3. Contribuții

În cazul proiectului de licență, Pixels editor, contribuțiile studentului sunt următoarele:

- documentarea studentului în legătură cu domeniul steganografiei (ascunderea de obiecte în alte obiecte pentru a putea fi transmise fără detecție);
- crearea unui plan și a unei strategii de realizare a proiectului ;
  - crearea meniului principal al aplicației;
  - divizarea meniului în submeniuri bine structurate;
  - implementarea pseudo-codului necesar;
- crearea unor exemple clare pentru o mai bună înțelegere a conceptului;
- căutarea și alegerea unei soluții optime;
- implementarea soluției;
- testarea soluției;

Pentru realizarea licenței am utilizat următoarele module:

- Python ca limbaj de programare cu interpretare de uz-general, interactiv, orientat pe obiecte și de nivel înalt;
- OpenCV ca bibliotecă de prelucrare a imaginilor;
- Numpy pentru lucrul cu matrici de dimensiuni mari;
- Pillow pentru crearea de noi imagini;
- Tinkter pentru crearea meniului, este un modul standard pentru Python în care se poate realiza interfața grafică a aplicației;
- Soundfile pentru lucrul cu fișiere audio;

## 4. Implementare

### 4.1. Crearea meniului

Meniul a fost creat cu ajutorul interfeței Tinkter din Python. Acesta este structurat în cinci mari module(Figure 1): ascunderea de imagini în imagini, ascunderea de text în imagini, ascunderea de imagini sau text în audio, ascunderea de imagini sau text în video și adăugarea de watermark în imagini și în video. Pagina principală constituie punctul de start în generarea sau preluarea obiectelor modificate în urma rulării comenzilor. Este realizat dintr-un frame care face legatura cu submeniurile fiecărui modul.

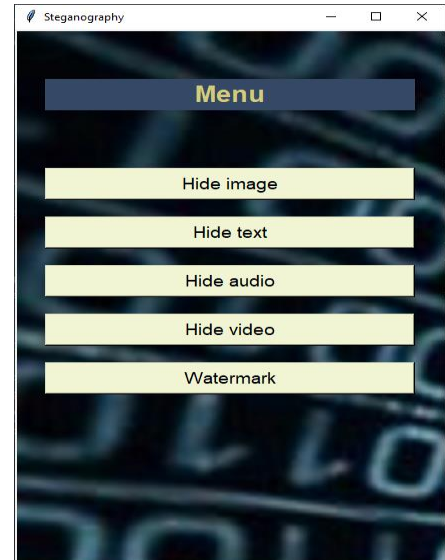


Figure 1-Meniul aplicației

#### 4.1.1. Hide image

Acest modul conține două căi de acces una către imaginea care urmează a fi modificată, imaginea gazdă (cover image), primul input box din frame și una către imaginea care se dorește a fi ascunsă (image to hide), al doilea input box din frame.

Totodată acest submeniu oferă utilizatorului posibilitatea de a naviga în propriul computer și de a alege o imagine fără a scrie un *path* către aceasta, calea de acces setându-se automat în momentul alegerii unei imagini. Pe lângă datele necesare pentru adăugarea unei imagini mai apare și un *scale* care lasă utilizatorului posibilitatea de a alege câți biți dintr-o imagine ar vrea să transmit mai departe.

În partea de jos a submeniului apar trei butoane: Load, Unload și Back to Home. Primul realizează încărcarea imaginii care dorim a fi ascunsă în imaginea gazdă. O imagine este realizată la nivel de bază dintr-o listă de pixeli care are valori cuprinse între 0 și 255. Pentru a putea realiza încărcarea datelor avem nevoie ca valoarea acestor pixeli să fie transformată în valoarea binară corespunzătoare. Acest proces este relativ simplu deoarece în imaginea copertă se modifică cei mai nesemnificativi n-biți cu cei mai semnificativi n-biți din imaginea care se dorește a fi transmisă

fară a fi observată. Cazul de bază în acest proces îl reprezintă modificarea ultimilor 4 biți din imaginea copertă cu primii 4 biți din imaginea care trebuie ascunsă(Figure 2).

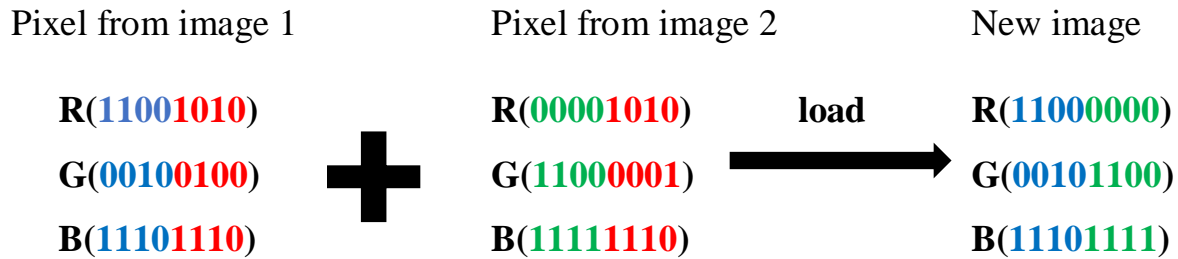
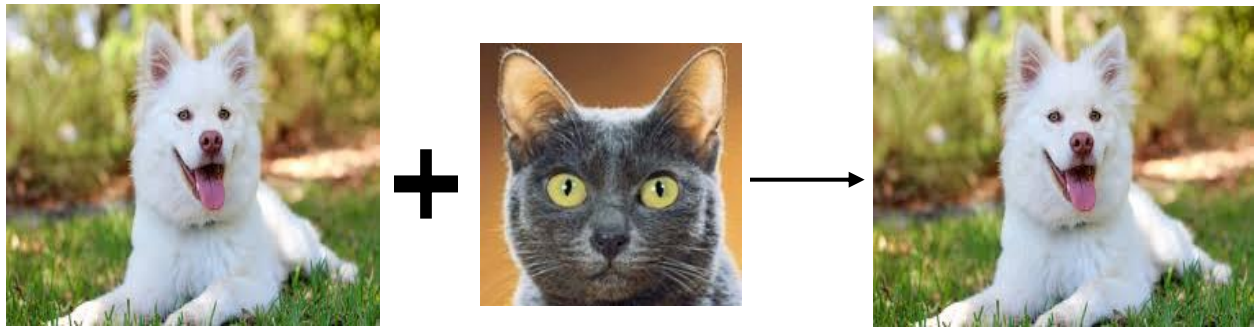


Figure 2-Modificarea biților



Al doilea buton este cel de Unload, este procesul invers, preluăm imaginea care nu doream a fi detectată din imaginea rezultată la pasul de Load. Reconstrucția imaginii transmise presupune extragerea celor mai semnificativi n-biți din imaginea modificată și adăugarea a 8-n biți pe care îi setăm cu 0. Imaginea rezultată va suferi cateva modificari din cauza faptului cei 8-n biți setați cu 0 nu concid cu cei 8-n biți pe care îi avea inițial. Ca în cazul de încărcare a imaginii cazul de baza presupune modificarea a 4 biți(Figure 3).

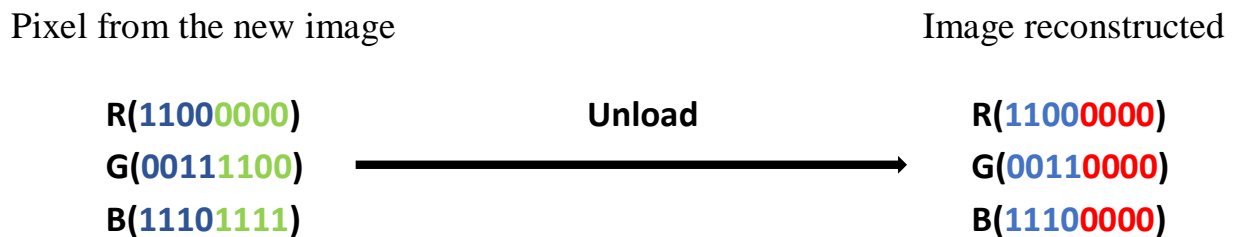


Figure 3-Reconstruirea imaginii



Al treilea buton este cel de Back to Home, acesta ne întoarce la pagina principala din meniu. Această operație se realizează prin apelul funcției ‘show\_page’ care face un salt (page.tkraise) la pagina la care am dori să ajungem.

### 4.1.2. Hide text

Acest modul conține alte două submodule Hide text invisible și Hide text visible dar și un buton de Back to Home. Hide text se ocupă cu ascunderea unui text în pixelii imaginii cover care urmează a fi modificată prin apelul funcției de încărcare prezentă în fiecare submodule. În acest modul găsim totodată și funcționalitatea de extragere a textului din imaginea returnată anterior.

#### 4.1.2.1. Hide text invisible

Primul submodule se numește *Hide text invisible* și după cum sugerează titlul această componentă se ocupă cu ascunderea textului în imagine fără ca acesta să poată fi detectat. Modulul de ascundere al textului invizibil conține un *input box* pentru adăugarea căii de acces a imaginii sau un buton de *browse* care permite alegerea unei imagini și automat completarea cu path-ul aferent acesteia. De asemenea conține un *text box* pentru inserarea textului care urmează a fi transmis mai departe prin intermediul imaginii, acest text vine însă cu o restricție, nu poate depăși 256 de caractere. Principalele butoane din acest modul sunt Load, Unload dar și Back care ne întoarce la meniul anterior.

Butonul de Load execută modificarea imaginii gazdă astfel încât textul să nu poată fi observant. Algoritmul se bazează pe modificarea valorii pixelilor din imagine în valoarea lor binară corespunzătoare. Totodată fiecare caracter din text este înlocuit cu codul sau ASCII pentru

ca ulterior acesta să fie transformat tot într-o valoare binară pe 8 biți care va fi plasată într-o listă. Încărcarea în imagine se realizează astfel: se parcurge imaginea pixel cu pixel și pe cel mai nesemnificativ bit din pixel se înlocuiește cu un bit din lista de biți(Figura 4).

- $a = \text{chr}(97) = 01100001$  în binar

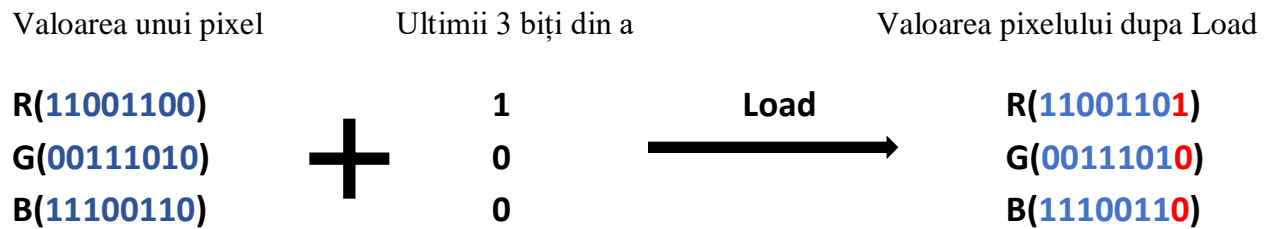


Figure 4-Încarcarea textului în imagine

Butonul de Unload parcurge imaginea nou creată pixel cu pixel și preia cel mai nesemnificativ bit. Lungimea textului care urmează a fi recreat se afla în primii trei pixeli pe ultimul bit. Dupa reconstruirea lungimii textului se va parcurge imaginea până când vom reuși să adunăm toți biții necesari pentru a recrea fiecare caracter, ulterior textul pe care am dorit să îl ascundem.

#### 4.1.2.2. Hide text visible

Al doilea submodule din modulul *Hide text* este *Hide text visible* care ascunde un text într-o imagine aleasă de utilizator. Diferența majoră dintre acest submodule și submoduleul anterior este: imaginea gazdă inițială și cea returnată după Load sunt similare. Pentru apelul funcțiilor de Load și Unload este necesară setarea căii de acces către imagine și adăugarea textului care se dorește a fi ascuns. Căutarea imaginii se poate realiza și prin apăsarea butonului de *Browse a file* care îi permit utilizatorului să aleagă o imagine din memoria calculatorului. Pentru ascunderea textului se execută comanda de *load\_text\_random* prin apsarea butonului de Load a cărei funcționalitate este:

- se parcurge textul dat și se transformă fiecare caracter în valoarea sa ASCII;
- se verifică dacă lungimea textului este divizibilă cu 3 pentru parcurgerea imaginii

- se parcurge imaginea și se plasează la poziții random valorile ASCII ale caracterelor, poziții care sunt plasate într-un dicționar partajat de utilizatori.

Pentru extragerea textului se setează calea de acces către imaginea modificată anterior după care se apasă butonul de Unload. Acest proces constă în parcurgerea dicționarului de poziții și preluarea fiecărei valori din fiecare celulă de pixeli. După ce aceste valori au fost extrase se crează din nou textul prin transformarea fiecărei valori ASCII în caracter.

### 4.1.3. Hide audio

*Hide audio* se ocupă cu ascunderea unui fișier audio într-o imagine, plasarea unei imagini sau a unui text într-un audio. Acest modul presupune crearea de noi fișiere wav(audio) care conțin informația care trebuie transferată. Este alcătuit din trei submodule:

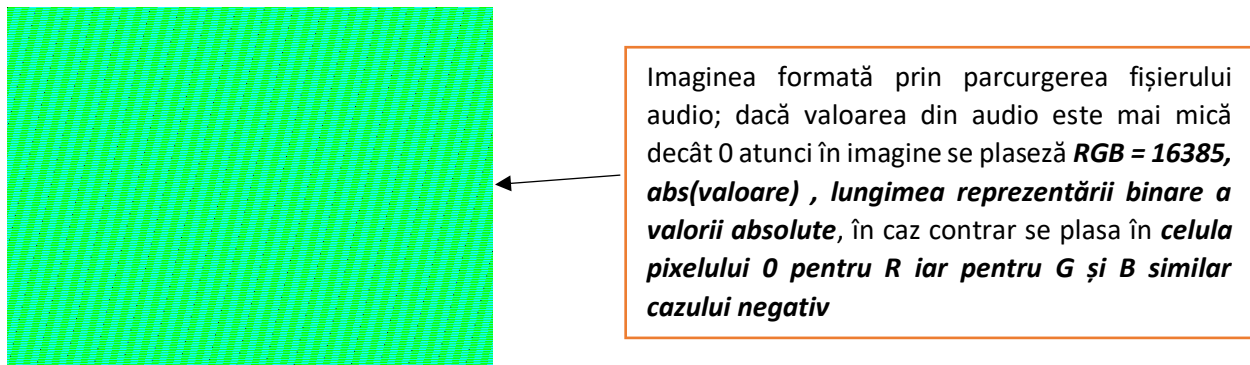


Figure 5-Creare imagine din audio

#### 4.1.3.1. Hide audio in image

Primul submodule este cel de ascundere de audio într-o imagine nou creată. Acesta conține un *Entry* pentru adăugarea căii de acces către fișierul audio sau un buton care permite alegerea unui fișier audio din cele existente și ulterior setarea path-ului. De asemenea mai conține două *radioButton*'s care ajută la crearea unei imagini gazdă unde va fi plasat fișierul wav. Pe lângă aceste date mai apare și un *Scale* cu valori cuprinse între 1 și 14 care permite utilizatorului să aleagă câți biți dorește să ascundă. Ascunderea fișierului se realizează prin apăsarea butonului de Load după setarea tuturor variabilelor. Această operație presupune:



- crearea unei imagini gazdă care să conțină valori ale pixelilor pe 16 biți (o imagine obișnuită are valori ale pixelilor pe 8 biți iar un fișier audio valori pe 16 biți, din această cauză s-a ales crearea unei noi imagini);
- crearea unei imagini din fișierul audio care presupune:
  - extragerea valorilor din audio;
  - setarea valorilor corespunzătoare pentru înălțime și lungime;
  - crearea unei imagini cu fundal negru de înălțimea și lungimea setate precedent;
  - parcurgerea acestei imagini și setarea fiecărui pixel cu valorile extrase din audio;
  - dacă valoarea este negativă pe pixel-ul R(roșu) apare 16384 (valoarea maximă admisă pe 16 biți), pe pixel-ul G(verde) apare valoarea absolută, iar pe pixel-ul B(albastru) apare lungimea binarului creat din valoarea absolută;
  - dacă valoarea este pozitivă pe pixel-ul R apare 0(valoarea minimă pe 16 biți), pe pixel-ul G valoarea audio iar pe pixel-ul B apare lungimea binarului creat din valoarea audio;
- se plasează valorile din imaginea audio creată într-o listă pentru a fi transformate în valorile binare corespunzătoare;
- în funcție de numărul de biți setați se modifică cei mai ne semnificativi biți din imaginea gazdă cu cei mai semnificativi biți din valorile audio;
- după plasarea tuturor valorilor audio în imagine se returnează noua imagine modificată;

Extragerea fișierului audio din imaginea creată presupune setarea path-ului corespunzător, de data aceasta către imagine nu către un audio, apoi apăsarea butonului de Unload care presupune parcurgerea imaginii gazdă doar între dimensiunile unde este plasat fișierul wav. Ulterior se preiau cei  $n$  biți și se adaugă valori 0 până la valoarea indicată de pixel-ul G. Aceste valori binare sunt transformate în valorile naturale(int) corespunzătoare, dacă prima valoare (a pixel-ului R) este 0 știm că este pozitivă, în caz contrar este o valoare negativă. Toate aceste valori extrase și modificate sunt plasate într-o listă care este dată ca parametru unei funcții pentru reconstrucția fișierului audio.

#### 4.1.3.2. Hide image in audio

Al doilea submodul este cel de ascundere a unei imagini într-un fișier audio. Pentru adăugarea imaginii și a fișierului audio utilizatorul are două posibilități, fie adaugă manual calea către acele fișiere după care trebuie să apese și butonul de *Set path* pentru a le face vizibile în program, fie apasă butonul *Browse a file* care îi permite să navigheze printre fișiere și să aleagă un obiect (setarea path-ului se face automat la alegerea obiectului).

Cele doua funcționalități prezente în acest submodule sunt cele de Load și Unload. Ascunderea imaginii presupune preluarea fiecărei valori din celula unui pixel, crearea unei imagini din fișierul audio în aceeași manieră în care a fost creată în modulul *Hide audio in image*. Se parcurge imaginea audio creată și la poziții random se modifică valorile RGB cu valori din lista de valori din imagine. Din imaginea audio nou creată se extrag toate valorile și se plasează într-o lista de valori pentru a reconstrui fișierul wav.

Pentru extragerea imaginii din audio se extrag pixelii de la pozițiile modificate anterior, se crează o imagine cu fundal negru de dimensiunile imaginii ascunse. Această imagine este parcursă și pentru fiecare celulă dintr-un pixel se setează o valoare din lista extrasă din audio.

#### 4.1.3.3. Hide text in audio

Al treilea submodul este cel de ascundere a unui text într-un fișier audio. *Hide text in audio* conține un *Entry* pentru setarea căii de acces către fișierul audio. Path-ul mai poate fi setat și prin apăsarea butonului de *Browse a File* care permite alegerea unui fișier wav dorit. De asemenea conține și un câmp pentru adăugarea textului care se dorește a fi ascuns. Ascunderea și extragerea se realizează prin apăsarea butoanelor de Load și Unload.

Butonul de Load presupune mai întâi setarea căii de acces și adăugarea textului, după ce acestea au fost setate din fișierul audio se crează o imagine prin apelul funcției *make\_new\_image* din submoduleul *Hide audio in image*. Ascunderea textului se realizează prin apelul funcției *load\_text\_random* din submoduleul *Hide text visible*, imaginea audio rezultată este parcursă pixel cu pixel, iar fiecare valoare este plasată într-o listă de valori care vor reconstrui fișierul audio.

Butonul de Unload presupune setarea căii de acces către fișierul audio modificat anterior. Din acest fișier se extrag valorile textului de la pozițiile random pe care ambii utilizatori le dețin, valorile numerice(int) extrase sunt de fapt valorile ASCII ale fiecărui caracter. După modificarea acestor valori în caracterul corespunzător se reconstruiește textul care se dorea a fi ascuns.

#### 4.1.4. Hide video

Hide video se ocupă cu ascunderea unei imagini sau a unui text într-un videoclip și este alcătuit din alte două submodule Hide image in video și Hide text in video. Acest modul presupune crearea unui nou videoclip după ce se modifică un frame.

##### 4.1.4.1. Hide image in video

Hide image in video este primul submodule și se ocupă cu ascunderea unei imagini într-un fișier video. Acesta conține două *input box*-uri pentru adăugarea căilor de acces pentru video sau imagini, ulterior trebuie să facem vizibil căile de acces în program prin apăsarea butonului de *Set Path*. O metodă alternativă pentru setarea manuală a path-urilor este apăsarea butonului de *Browse a file* care permite alegerea unui fișier din computer. De asemenea mai există un scale care permite setarea numărului de biți care dorim să fie ascunși. Un videoclip este alcătuit din mai multe imagini care sunt derulate la un anumit timp. Ascunderea imaginii presupune setarea path-urilor și a numărului de biți, apoi apăsarea butonului de Load care apelează funcția *final\_load()*. Această funcție realizează următoarele funcționalități:

- se calculează numărul de frame-uri din videoclip-ul dat ca parametru;
- se inițializează valoarea parametrului *random\_frame* cu o valoare random între 0 și numărul de frame-uri;
- se extrage frame-ul de la poziția random setată anterior;
- se apelează funcția *load* din submoduleul *Hide image in image* care are ca parametri frame-ul din video (este o imagine de la un anumit timp), imaginea setată și numărul de biți;
- rezultatul apelului este o nouă imagine care conține imaginea care se dorește să fie transmisă este plasată în video la poziția random;
- se afișează videoclipul nou creat;

Extragerea imaginii din video presupune adăugarea parametrilor necesari pentru apelul funcției de *unload\_image\_from\_video()*. Această funcție deschide videoclipul dat ca parametru (videoclipul creat la pasul anterior), extrage frame-ul de la poziția random setată la apelul Load apoi se apează funcția *unload\_show* din submodulul *Hide image in image*. Imaginea rezultată după acest apel este imaginea care s-a dorit a fi transmisă.

#### 4.1.4.2. Hide text in video

Hide text in video este al doilea submodule din modulul de ascundere a obiectelor în videoclip și realizează ascunderea unui text într-un video. Acesta conține un *Entry* pentru adăugarea căii de acces către fișierul video, această cale mai poate fi adăugată prin apăsarea butonului de *Browse a file* care permite alegerea unui fișier mp3 sau avi dintre cele existente în memorie. De asemenea mai apare un câmp pentru inserarea textului care se dorește a fi ascuns. Ascunderea textului se realizează astfel: se setează parametrii necesari, se apasă butonul Load, se extrage un frame din videoclip de la o poziție random, se modifică acel frame prin apelul funcției *load\_text\_random* din submodulul *Hide image visible*. Imaginea modificată este plasată la poziția random în videoclip după care se salvează și afișează noul videoclip.

Extragerea textului presupune setarea căii de acces către videoclipul modificat anterior, deschiderea acestuia și extragerea frame-ului în care se află textul. Funcția Load returnează totodată și un dicționar care conține pozițiile din frame care au fost modificate. De la aceste poziții se preiau valorile pixelilor și se modifică valorile în caracterul corespunzător pentru reconstrucția textului.

#### 4.1.5. Watermark

Modulul de Watermark este creat pentru adăugarea și eliminarea de logo în/din imagine sau video. Watermark conține alte 4 submodule: *Add image logo to image*, *Add text logo to image*, *Add image logo to video* și *Add text logo in video*.

#### 4.1.5.1. Add image logo to image

*Add image logo to image* este primul submodul și se ocupă cu adăugarea unui logo creat dintr-o imagine aleasă de utilizator la o imagine gazdă. Acest submodul conține două *input box*-uri pentru adăugarea căilor de acces către imaginea care urmează a fi modificată și către imaginea logo. De asemenea mai conține cinci *RadioButton*'s pentru setarea poziție logo-ului și un *Scale* pentru setarea parametrului alpha (gradul de vizibilitate a imaginii logo).

Adăugarea unui logo (Figure 6) se realizează prin apăsarea butonului de Load după setarea tuturor parametrilor. Acest buton apelează funcția *add\_watermark\_img()* care conține alte patru funcții:

- *create\_mask()* -> preia imaginea logo, crează o nouă imagine gri din aceasta apoi apelează funcția *threshold* din OpenCV care returnează o imagine în care partea alba reprezintă obiectul iar partea neagră reprezintă fundalul;
- *remove\_background()* -> preia imaginea logo și imaginea creată la pasul anterior, crează o noua imagine cu fundalul negru;
- *create\_invisible\_image* -> preia imaginea creată anterior și prin apelul funcțiilor *threshold*, *split*, *merge* din OpenCV se crează o nouă imagine fără background;
- *add\_logo()* -> se adaugă imaginea logo creată la pasul anterior peste imaginea gazdă prin apelul funcției OpenCV *addWeighted*.

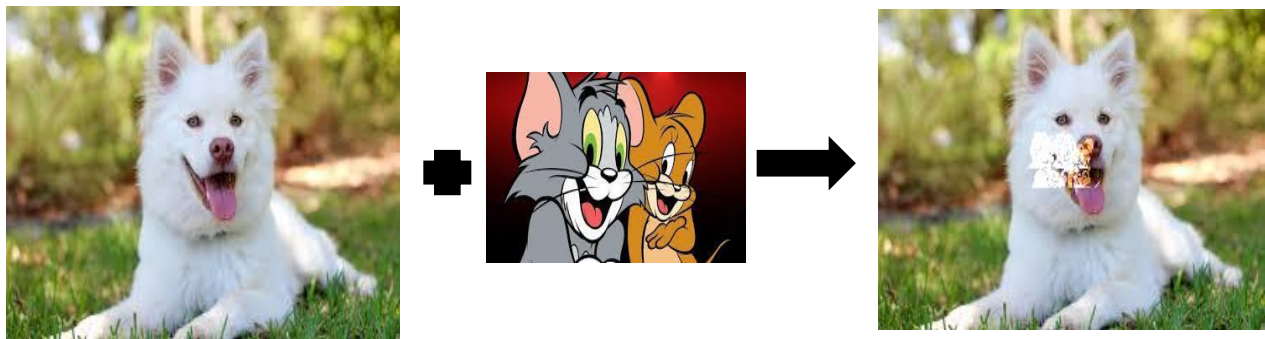


Figure 6 – Adăugare imagine logo

Eliminarea unui logo (Figure 6) se realizează prin apăsarea butonului Unload după ce a fost adăugată calea de acces către imaginea modificată la pasul de Load. Mai întâi se crează o mască, o imagine cu fundal negru de aceeași dimensiune ca imaginea gazdă care la pozițiile setate la Load va avea masca creată la apelul funcției *create\_mask()*. Cu ajutorul funcției *inpaint* din OpenCV în locul unde se află imaginea logo se încearcă reconstrucția imaginii inițiale prin preluarea valorilor pixelilor din apropiere sau prin blurare.

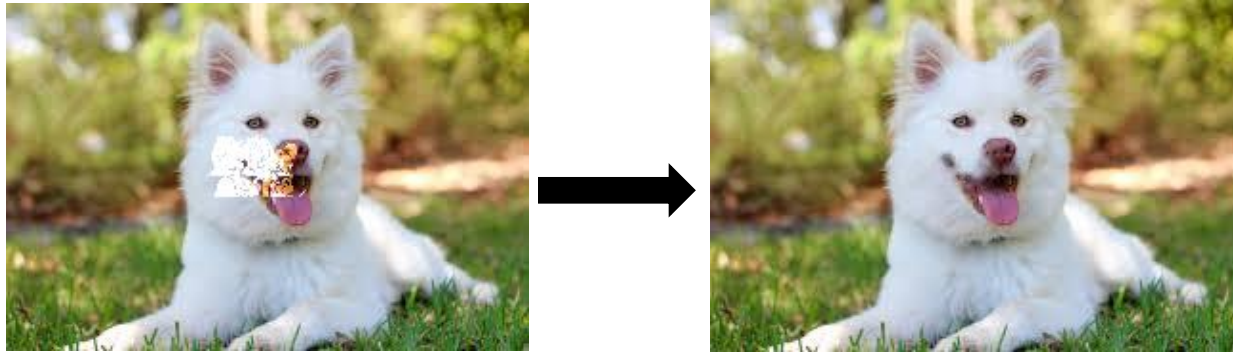


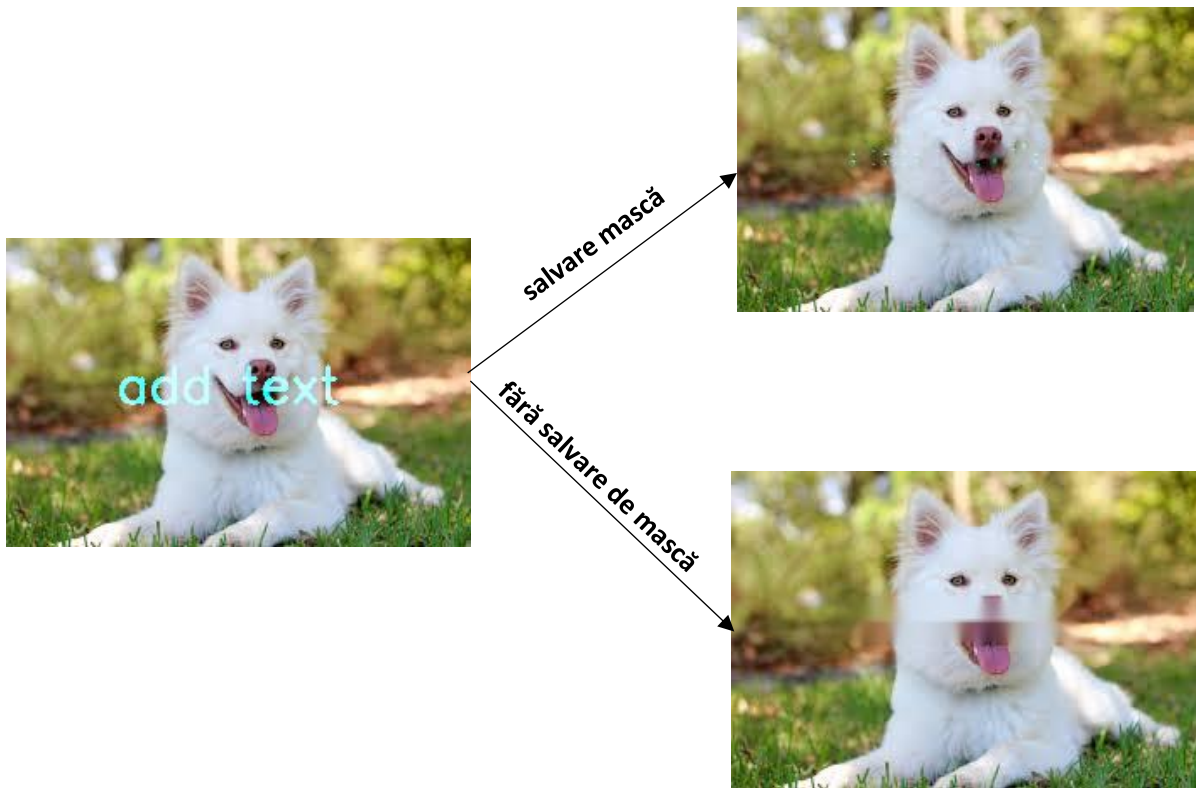
Figure 7- Eliminare imagine logo

#### 4.1.5.2. Add text logo to image

*Add text logo to image* este al doilea submodul, acesta se ocupă cu adăugarea unui text ca logo într-o imagine. Este alcătuit dintr-un *input box* pentru adăugarea path-ului către imaginea gazdă, un câmp pentru adăugarea textului, un câmp pentru setarea culorii textului, cinci *RadioButton*'s pentru setarea poziției, alte două *RadioButton*'s pentru a salva sau nu masca textului (dacă salvăm masca se va crea o imagine cu fundal negru și cu textul logo alb, în caz contrar doar imaginea cu fundal negru de dimensiunile corespunzătoare) și un *Scale* pentru setarea nivelului de vizibilitate al textului, parametrul *alpha*.

Adăugarea textului logo presupune crearea unei imagini, a unei măști, care va avea dimensiunile proporționale cu lungimea textului, va avea fundalul negru și prin apelul funcției OpenCV *putText* i se va adăuga peste textul dat ca parametru cu culoarea specificată. Apoi se apelează funcțiile *create\_mask*, *remove\_background*, *create\_invisible\_image* din submodulul *Add image logo to image* pentru crearea măștii finale care prin apelul funcției *add\_logo\_text* va fi adăugată peste imaginea gazdă la poziția dată și cu nivelul de vizibilitate setat.

Eliminarea textului logo presupune preluarea măştii creată prin apelul funcţiei *create\_txt\_image*, parcurgerea acesteia şi extragerea fiecărei valori din celulele pixelilor într-o listă de valori apoi crearea unei imagini cu fundal negru de dimensiunile imaginii gazdă şi plasarea la poziţiile setate la Load a valorilor din lista creată anterior. După ce această imagine a fost creată se apelează funcţia *inpaint* care va încerca eliminarea logo-ului. Dacă la pasul de load am ales să salvăm masca extragerea textului va fi mai puţin vizibilă întrucât în acest caz se încearcă doar reconstrucţia pixelilor unde se afla textul. Dacă am ales să nu salvăm textul reconstrucţia pixelilor se va face pe întreaga poţiune unde se afla imaginea creată din textul dat.



#### 4.1.5.3. Add image logo to video

*Add image logo to video* este al treilea submodul, acesta se ocupă cu adăugarea unei imagini logo la un videoclip. Submodulul conţine două input box-uri, pentru adăugarea videoclipului şi pentru adăugarea imaginii, *Radio Button*'s pentru selectarea poziţiei unde va fi plasat logo-ul şi un *Scale* pentru setarea nivelului de vizibilitate a imaginii logo.



Inserarea imaginii logo presupune setarea tuturor parametrelor, apăsarea butonului de Load care apelează funcția *final\_load*. Aceasta funcție este împărțită în doi pași importanți:

- apelul funcției *compute\_logo\_image* presupune apelul metodelor din submodulul *Add image logo to image* pentru a crea viitoarea imagine logo fără background;
- apelul procedurii *add\_logo\_video\_image* presupune parcurgerea videoclipului frame cu frame și crearea unei măști care va fi suprapusă peste fiecare frame.

Extragerea imaginii logo necesită setarea căii de acces către videoclipul creat la pasul de Load apoi apăsarea butonului de Unload. Procedura apelată la extragerea logo-ului este *remove\_logo\_video\_image* care cât timp videoclipul este deschis crează masca pentru extragerea logo-ului prin apelul metodei *compute\_mask* din submodulul *Add image logo to image*, apoi se încearcă eliminarea logo-ului de la fiecare frame.

#### 4.1.5.4. Add text logo to video

*Add text logo image* este ultimul submodul și se ocupă cu adăugarea unui text logo la un videoclip. Acest submodule conține un *input box* pentru adăugarea căii de acces către videoclip, un câmp pentru adăugarea textului, un câmp pentru setarea culorii textului, *Radio Button's* pentru setarea poziției și pentru salvarea măștii care va ajuta la extragerea logo-ului și un *scale* pentru setarea parametrului alpha.

Adăugarea textului logo în videoclip se realizează similar cu adăugarea textului logo în imagine, singura diferență este ca logo-ul se va adăuga la fiecare frame al videoclipului, un frame fiind o imagine care apare la un anumit timp din videoclip.

Extragerea textului logo din videoclip presupune adăugarea căii de acces către videoclipul creat anterior, parcurgerea videoclipului frame cu frame, crearea unei măști pentru extragerea logo-ului apoi apelarea funcțiilor OpenCV necesare. În funcție de alegerea făcută la pasul de adăugare a logo-ului extragerea poate fi mai mult sau mai puțin vizibilă întru-cât dacă am ales să salvăm masca se vor modifica doar pixelii unde se afla scrisul, în caz contrar se va modifica întreaga regiune în care se afla imaginea creată din text.



## 5. Testare

Testarea presupune verificarea precondițiilor pentru Load și Unload pentru ca programul să ruleze corespunzător și să furnizeze răspunsul corect.

Pentru primul modul în cazul apelului Load s-au folosit verificările următoare (cazul Unload conține doar verificările de existență și de setare al path-urilor):



Modulul de ascundere a unui text în imagine presupune pe lângă verificarea existenței și setării imaginii și următoarea verificare (cazul Unload presupune doar verificarea existenței și setării path-ului imaginii modificate):

```
if path1 == "" or text == "":
    messagebox.showinfo("Info", 'Incorrect input.\nThe input is not set.')
```

se verifică dacă a fost adăugat text

Modulul Hide audio presupune pe lângă setarea și verificarea căilor de acces către imagine/audio sau adăugarea textului următoarele verificări în cazul Hide audio in image:

se verifică dacă a fost setat  
radio button-ul

se verifică numărul de biți

```
if path1 == "" or v.get() not in [1, 2] or (bite < 1 or bite > 14):
    messagebox.showinfo("Info", 'Incorrect input.\nThe input is not set.')
else:
    if os.path.exists(path1) is False:
        messagebox.showinfo("Error", 'Audio path does not exists!')
        self.path_one.delete(0, 'end')
    else:
        final_load_audio(bite, path1, v.get())
        self.path_one.delete(0, 'end')
```

Modulul Hide video presupune verificări similare cu cele din modulele Hide image și Hide text. Singura diferență este că în acest submodule se adaugă o cale de acces către un videoclip iar în cazul ascunderii de imagine se va verifica dimensiunea imaginii în comparație cu dimensiunea unui frame din videoclip.

Modulul de Watermark presupune pe langă verificările existenței și setării path-urilor către imagine/ video și adăugarea textului următoarele examinări:

se verifică dacă a fost setată poziția de inserare a logo-ului

se verifică dacă parametrul alpha este corect

```
if path1 == "" or path2 == "" or v.get() not in [1, 2, 3, 4, 5] or (bite < 0 or bite > 1):
    messagebox.showinfo("Info", 'Incorrect input.\nThe input is not set.')
```

se verifică dacă imaginea logo este mai mică decât imaginea copertă

```
if row_logo//5 > row_cover or col_logo//5 > col_cover:
    self.path_one.delete(0, 'end')
    self.path_two.delete(0, 'end')
    messagebox.showinfo("Message", 'Logo image should not be larger than cover image!')
```

se crează un pattern de adăugare a culorii textului logo apoi se verifică dacă pattern-ul a fost respectat

dacă nu a fost găsită o potrivire afisează un mesaj

```
pattern = r"(\d+),(\d+),(\d+)"
match = re.search(pattern, color)
if match is None:
    self.color.delete(0, 'end')
    messagebox.showinfo("Info", 'Incorrect input.\nThe input should be integer values.')
else:
    search = re.search(pattern, color, re.IGNORECASE)
    if int(search.group(1)) > 255 or int(search.group(2)) > 255 or int(search.group(3)) > 255:
        self.color.delete(0, 'end')
        messagebox.showinfo("Info", 'Values of R, G, B should be between 0 and 255!')
```

dacă se respectă pattern-ul dar numerele inserate sunt mai mari decât 255 atunci se cere reinserarea acestora

## 6. Concluzii

În concluzie sunt de părere că Pixels editor este o aplicație excelentă în ceea ce privește ascunderea de imagini, text, audio în diferite medii vizuale. Posibilitatea de a avea o gamă largă din care să poți alege ce vrei să ascunzi și unde să ascunzi este un atu fața de alte aplicații similare care nu oferă decât o funcționalitate pe rând.

Procesarea de imagini este importantă pentru ascunderea unui mesaj într-un alt obiect fără ca alți utilizatori să observe existența acestuia, doar expeditorul și receptorul știu de existența sa. Spre deosebire de steganografie ascunderea de imagini prin criptografie este mult mai vizibilă pentru celelalte persoane și automat mai predispusă să fie piratată.

Spre deosebire de alte aplicații asemănătoare, Pixels editor, oferă un nivel mult mai mare de libertate. În special, modul în care utilizatorul poate alege ce fișier să ascundă, câți pixeli să se modifice, ce tip de audio să adauge, la ce poziție să adauge un logo, dacă să salveze sau nu masca unui logo, ce culoare să aibă logo-ul și nivelul de vizibilitate a logo-ului.

Ca îmbunătățiri și direcții viitoare, m-am gândit la crearea unui algoritm mai eficient de ascundere a unei imagini în imagini, să verific câți pixeli au fost modificați în funcție de modul de salvare a unei imagini png/jpg sau a unui video mp4/avi. De asemenea o altă îmbunătățire ar fi plasarea aplicației în mediul online, salvarea fișierelor într-o bază de date și citirea textului dintr-un fișier dat de către utilizator.

## 7. Bibliografie

- “*Digital Watermarking and Steganography*” by Ingemar J. Cox, Matthew L. Bloom;
- “*Information Hiding Techniques for Steganography and Digital Watermarking*” by Stefan Katzenbeisser and Fabien A. P. Petitcolas;
- “*Disappearing Cryptography*” by PeterWayner
- <https://towardsdatascience.com/steganography-hiding-an-image-inside-another-77ca66b2acb1>
- <https://null-byte.wonderhowto.com/how-to/steganography-hide-secret-data-inside-image-audio-file-seconds-0180936/>
- <https://stackoverflow.com/>
- <https://docs.python.org/3/library/tkinter.html>
- <https://pysoundfile.readthedocs.io/en/latest/>
- [https://pywavelets.readthedocs.io/en/latest/ref/2d-decompositions-overview.html?fbclid=IwAR3vansijHCq2Erz5uUsnmi9\\_2bovgq7NFXBOT2DeqFh0eML9RZp8rSlnIE](https://pywavelets.readthedocs.io/en/latest/ref/2d-decompositions-overview.html?fbclid=IwAR3vansijHCq2Erz5uUsnmi9_2bovgq7NFXBOT2DeqFh0eML9RZp8rSlnIE)
- [https://www.learnopencv.com/read-write-and-display-a-video-using-opencv-cpp-python/?fbclid=IwAR3YaYxcRobGLSgV8HrKMCL\\_0U-INU4\\_tF\\_3pOhJlzkvtv1Nq8\\_1e89Qgcxw](https://www.learnopencv.com/read-write-and-display-a-video-using-opencv-cpp-python/?fbclid=IwAR3YaYxcRobGLSgV8HrKMCL_0U-INU4_tF_3pOhJlzkvtv1Nq8_1e89Qgcxw)
- <https://searchsecurity.techtarget.com/definition/steganography?fbclid=IwAR1YRMAzfxY8ZtJSMrplA1aWrI-dHMoBznjnzMJhfEE4dlrO05J1Giab-VA>
- [https://github.com/intel-iot-devkit/Video-Analytics-OpenCV/tree/master/tutorials/opencv/Python/sample\\_09\\_DOG\\_video?fbclid=IwAR3vansijHCq2Erz5uUsnmi9\\_2bovgq7NFXBOT2DeqFh0eML9RZp8rSlnIE](https://github.com/intel-iot-devkit/Video-Analytics-OpenCV/tree/master/tutorials/opencv/Python/sample_09_DOG_video?fbclid=IwAR3vansijHCq2Erz5uUsnmi9_2bovgq7NFXBOT2DeqFh0eML9RZp8rSlnIE)
- [https://docs.opencv.org/master/d0/d86/tutorial\\_py\\_image\\_arithmetics.html?fbclid=IwAR0QPOY2TI0qZ7I8Ojg45DfSBPp8BLpU5hgd588Uq4m-Xkflf1dNdSMa-Rg](https://docs.opencv.org/master/d0/d86/tutorial_py_image_arithmetics.html?fbclid=IwAR0QPOY2TI0qZ7I8Ojg45DfSBPp8BLpU5hgd588Uq4m-Xkflf1dNdSMa-Rg)